UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE GRADO

Librería para Interacción Natural empleando técnicas de Visión por Computador

Pablo Manuel Caballero Chaparro

Tutor: Francisco Jurado Monroy

Ponente: Rosa M. Carro Salas

Julio 2014

Resumen

A lo largo de este trabajo se desarrollará una librería en Java que permitirá a la aplicación que la emplee realizar una Interacción Natural con el usuario. Esta se valdrá de la librería OpenCV para utilizar técnicas de visión por computador sobre las imágenes obtenidas a través de una cámara y así interpretarlas detectando gestos realizados por el usuario. Tras esta detección la librería ejecutará una función de la aplicación que la utiliza. Será el desarrollador de la aplicación el que determinará qué función ejecutar tras la detección de cada uno de los gestos soportados por la librería. Por otro lado, la librería contará con un modo que permitirá al usuario desplazar el puntero del ratón sobre una serie de componentes de la aplicación empleando los mismos gestos, así como hacer click sobre ellos. Será de nuevo el desarrollador el que facilitará este conjunto de componentes. También decidirá qué modo de los disponibles utilizar en cada momento.

Para el desarrollo de la librería será necesario seleccionar adecuadamente los gestos soportados por la misma, que estarán recogidos en un protocolo de interacción. Además, se desarrollará una galería de imágenes en Java que empleará esta librería de Interacción Natural para poder ilustrar y probar su uso.

Abstract

Throughout this work a library in Java that will allow the application which uses it to perform a natural user interaction will be developed. It will apply the OpenCV library to use computer vision techniques over the images obtained through a camera, detecting and interpreting gestures made by the user. After the gesture detection, the library will execute a function of the application which controls it. The application's developer will have the ability to determine which function must be executed after detecting each of the gestures supported by the library. In addition, the library will have a way to allow the user to move the mouse over a group of the application's components using the same gestures and to click on them too. The developer will provide this set of components to the library. Also, he will decide which mode is available at every moment.

To achieve a correct result, it will be necessary to properly select the gestures supported by it, which will be collected in an interaction protocol. Furthermore, a gallery of images in Java will be developed using this natural interaction library to illustrate and test it.

Palabras Clave

Trabajo Fin de Grado (TFG)

Interacción Natural

Visión por computador

Librería

OpenCV

Java

Galería de imágenes

Key Words

Final Project

Natural Interaction

Computer Vision techniques

Library

OpenCV

Java

Images Gallery

Índice General

Índice de Figuras

1. Introducción

El paradigma de la Interacción Natural permite diseñar Interfaces Naturales de Usuario. Estas se valen de los movimientos gestuales realizados por el usuario permitiéndole interactuar con el sistema sin la necesidad de emplear dispositivos de entrada como el ratón, el teclado o un joystick. De esta manera, es el propio cuerpo o la voz del usuario lo que es interpretado como una entrada al sistema y permite interactuar con el mismo.

Dentro del campo de la Interacción Natural, entre otras, se emplearán técnicas de Visión por Computador, las cuales permiten al sistema extraer y analizar propiedades del mundo real a partir de un conjunto de imágenes obtenidas mediante una cámara. Gracias a estas técnicas se podrán localizar las propiedades que permitan identificar los movimientos del usuario y hacer así que la aplicación actúe en función de estos movimientos detectados.

1.1. Motivación del trabajo

La principal motivación para llevar este trabajo a cabo consiste en poder aportar alternativas a las clásicas interacciones del usuario con el computador a partir de dispositivos de entrada como típicamente son el teclado y el ratón. Es por ello que no necesitará valerse de ningún dispositivo hardware como los mencionados. Por el contrario, utilizará sus propias manos o un objeto sencillo para indicarle al sistema qué es lo que desea hacer.

Estas alternativas serán más intuitivas ya que los movimientos, gestos o acciones que el usuario realice frente al ordenador se corresponderán de forma lógica con funcionalidades del sistema. Aunque el usuario necesitará conocer y aprender previamente el conjunto de gestos/funcionalidades soportados, será más intuitivo pues el ser humano es social por naturaleza, y se comunica no solo de manera verbal, si no que lo hace también de forma escrita e incluso con lenguaje de signos. Pero además, por ejemplo al charlar con alguien, no nos comunicamos tan solo de manera verbal, siempre acompañamos nuestro discurso con gestos que ayudan al oyente a interpretar mejor aquello que decimos o, mejor, que queremos decir. Tanto es así que las personas suelen emplear gestos que se asocian con sentimientos o estados de humor y pueden ser reconocidos por un pequeño grupo de amigos, por todo un colectivo (cultural, nacional, religioso…), o incluso a nivel mundial. Esta forma de expresión mediante gestos es parte de nuestra forma de comunicarnos y podrían emplearse perfectamente para la comunicación con un sistema informático.

Por otro lado, el hecho de trabajar en el área multimedia, es otra de las motivaciones para elaborar este trabajo. A lo largo de la carrera no he tocado este campo al no elegir ninguna de las optativas que lo tratan y me gustaría obtener conocimientos en el mismo.

1.2. Objetivos del proyecto

En el presente Trabajo Final de Grado se plantean varios objetivos para lograr que el proyecto concluya con éxito. En primer lugar habrá que implementar una librería que permita a la aplicación que la utilice realizar una Interacción Natural con el usuario. Esta empleará técnicas de visión por computador para capturar y reconocer las acciones del usuario haciendo uso de la librería OpenCV. Se analizarán los fotogramas obtenidos a través de una cámara para saber qué está haciendo el usuario frente a esta.

Por otro lado, para que este análisis sea posible, se creará un protocolo de interacción, compuesto por diferentes gestos o movimientos, que el usuario deberá de conocer y realizar para ser reconocidos.

La librería detectará las acciones del usuario y dependerá del programa principal interpretar estos eventos a su antojo. Esto permitirá la portabilidad de la librería a cualquier programa, que podrá soportar todos o un subconjunto de los gestos disponibles y enlazarlos con sus funcionalidades.

Por último, para probar el correcto funcionamiento de la librería se implementará un programa que haga uso de ella y que se servirá del protocolo de interacción para que el usuario pueda hacer uso de sus funcionalidades. El programa se tratará más concretamente de una galería de imágenes.

De manera resumida los objetivos del proyecto son:

Objetivo 1. Crear un protocolo de interacción compuesto por un conjunto reducido de gestos o acciones simples que sea portable a cualquier aplicación.

Objetivo 2. Crear una librería que de acuerdo con el protocolo anterior sea capaz de identificar cualquiera de los gestos definidos y actuar en consecuencia. Deberá ser altamente portable para su ejecución en diferentes máquinas.

Objetivo 3. Demostrar y probar las capacidades de la librería desarrollada junto con otro programa que haga uso de sus funcionalidades.

1.3. Estructura del Proyecto

El resto del documento se estructurará como sigue:

- **Capítulo 2**, Estado del Arte:

  En este capítulo se verán las posibilidades de la Interacción Natural, haciendo hincapié en la visión por computador. Para ello se hablará sobre proyectos actuales en este campo. Se prestará especial atención a aquellos trabajos que empleen la librería OpenCV para aplicar estas técnicas de visión por computador en sus proyectos.

- **Capítulo 3**, Diseño:

  Dentro de este capítulo se va a especificar la manera en la que se ha afrontado el proyecto para poder cumplir con los objetivos definidos. Se explicará el ciclo de vida del software elegido, viendo cómo se ha escogido el mejor modelo para desarrollar el proyecto de acuerdo con sus requerimientos. Se mostrará la planificación del proyecto, aplicando el modelo de desarrollo escogido, definiendo el orden en el que se ha resuelto y acotando las tareas en el tiempo. También se hablará sobre las tecnologías seleccionadas para la mejor resolución de los objetivos y teniéndolas en cuenta se mostrará el diseño de cómo será la arquitectura final del sistema

- **Capítulo 4**, Desarrollo:

  En este capítulo se verá cómo se han implementado cada uno de los elementos que componen nuestro sistema. Se planteará cual deberá ser el conjunto de gestos soportados por el protocolo de interacción y se especificarán las acciones necesarias para la detección de los mismos. Seguidamente se hablará sobre las decisiones tomadas para la implementación de la librería de Interacción Natural así de cómo esta se ha llevado a cabo. También se explicará cómo se ha incluido en ella el protocolo de interacción anteriormente presentado para detectar el conjunto de gestos.

- **Capítulo 5**, Evaluación de la librería de Interacción Natural:

  Llegados a este capítulo se probarán y mostrarán las funcionalidades implementadas en nuestra librería de Interacción Natural a través de una aplicación de ejemplo, una galería de imágenes. En primer lugar se presentará esta aplicación para así familiarizarnos con ella y conocer las funcionalidades de las que dispone. A continuación se mostrará como se ha implantado la librería desarrollada a la galería de imágenes y se hablará de las posibilidades existentes para el uso de la librería en distintos escenarios.

- **Capítulo 6**, Conclusiones:

  En el último capítulo se expondrán los resultados obtenidos tras la elaboración del proyecto. También se hablará de posibles mejoras y se valorará de manera personal la experiencia al desarrollarlo.

2. Estado del arte

En el presente capítulo se abordará el estado del arte, prestando especial atención a aquellos trabajos que emplean técnicas de visión por computador. Se hablará de proyectos actuales en este campo, y en especial de aquellos trabajos que emplean la librería OpenCV para aplicar técnicas de visión por computador en sus proyectos.

2.1. Dispositivos para Interacción Natural

Para poder diseñar Interfaces Naturales en el ámbito de la Interacción Natural, existen diferentes dispositivos hardware con distintos tipos de tecnologías incorporadas que permiten realizar un reconocimiento más o menos fidedigno de la realidad que se sitúa frente a ellos. A continuación se hablará sobre algunos de estos dispositivos:

-Structure Sensor [2]: Mediante este sensor se puede obtener una representación en 3D de todo lo situado frente a él. De este modo es capaz de introducir objetos de realidad aumentada sobre el escenario o crear modelos en 3D de los objetos que lo componen. También permite obtener las medidas de estos objetos así como las de habitaciones enteras.

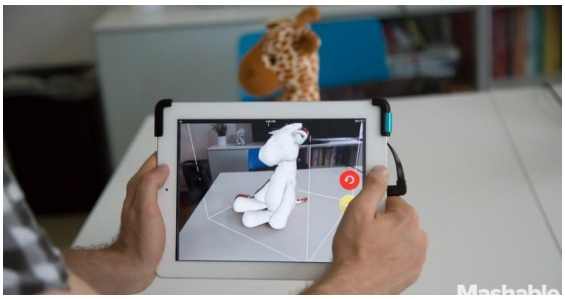

Ilustración I: Structure Sensor, imagen extraida de [19]

-Kinect [3]: Es un dispositivo de Microsoft, diseñado en su origen por la compañía Primense, principalmente desarrollado para la consola Xbox-360 que también es compatible con Windows 7 y 8. Está compuesto por una cámara RGV, un sensor de profundidad y un micrófono de múltiples matrices que le permiten capturar el movimiento de todo el cuerpo del usuario en tres dimensiones y realizar reconocimiento facial y de voz.

Ilustración II: Kinect, imagen extraida de [20]

-Leap Motion Controller [1]: Este dispositivo permite llevar un seguimiento de la posición de las dos manos del usuario así como de los diez dedos que las componen. Está equipado por una serie de luces LED y cámaras que le permiten escanear la zona que se encuentra sobre este.

Ilustración III: Leap Motion Controller, imagen extraida de [21]

-PlayStation Move [4]: Este dispositivo fue desarrollado por Sony como controlador de juegos para su videoconsola PlayStation 3. A parte de incluir distintos sensores de movimiento, de manera análoga al WiiMote de Nintendo [23], incluye una esfera en su extremo que se ilumina. Esta esfera, junto con la cámara de la misma videoconsola PlayStation Eye, permite al sistema hacer un reconocimiento de los movimientos del PlayStation Move. Se realiza un reconocimiento por color y es por ello que la esfera es capaz de cambiar de color para que este no coincida con ninguno de los que se encuentren en la habitación.

Ilustración IV: PlayStation Move y PlayStation Eye, imagen extraida de [22]

-Amazon Fire Phone [5]: Se trata de un teléfono móvil de Amazon, el cual incluye cuatro cámaras de bajo consumo en su cara frontal que le permiten responder a la manera en la que el usuario sujeta, mira o mueve el teléfono.

Ilustración V: Amazon Fire Phone, imagen extraida de [5]

Todos estos dispositivos emplean tecnologías que en conjunto permiten grandes resultados en las tareas de Visión por Computador para las que fueron diseñados. Todos ellos tienen un coste muy superior al de una cámara web, tanto por el esfuerzo empleado en su desarrollo como por el precio de sus componentes. Es por ello que en este trabajo se utilizara una cámara web como dispositivo de captura de imágenes. Para procesar la información capturada por la cámara web y convertirla en eventos una vez identificados los correspondientes gestos, se empleará una librería que permita procesamiento de imágenes. En particular se utilizará la librería OpenCV que, siendo de código abierto, nos permite aplicar técnicas de visión por computador a la entrada obtenida de la cámara web para permitir reconocimiento facial, identificación de colores o seguimiento de objetos, técnicas que también son aplicables con los dispositivos anteriormente comentados. Los detalles de esta librería se darán en la sección 3.3 donde se presentarán las tecnologías a emplear. Sin embargo, antes de adentrarnos en las características de la librería, se mostrarán en la siguiente sección aquellos trabajos asociados con el estado del arte que la emplean.

2.2. Interacción Natural empleando OpenCV

Existen muchos proyectos en este campo, que se sirven de la potencia y sencillez de la librería OpenCV para reconocer una gran variedad de gestos. La mayoría de los desarrollados hasta la fecha que la emplean están escritos en C++.

Existe un trabajo muy interesante [12] en el que se desarrollan tres proyectos empleando técnicas de visión por computador. Uno de ellos relaciona cada gesto con una nota del piano, permitiendo así tocar una melodía. Otro permite seguir la mano del usuario para controlar la posición del ratón del sistema operativo y servirse de diversos gestos para hacer click o scroll. Y por ultimo un proyecto que permite detectar diversos gestos que controlen un sencillo videojuego de lucha.

Otro de los trabajos encontrados [11] reconoce en los gestos de las manos las letras del abecedario según el lenguaje de signos.

Hay un trabajo [13] que mediante una serie de gestos permiten al usuario comunicarse con el sistema operativo. De esta manera el usuario puede lanzar el reproductor de videos, pausarlo, reanudarlo o incluso bajar o subir el volumen del audio.

Existen proyectos que utilizan la detección de objetos externos a la mano del usuario, como puede ser un puntero o una pelota [18]. De esta manera es el objeto clave lo que se reconoce y puede utilizarse para llevar una acción a cabo.

Todos estos proyectos utilizan el mismo principio de detectar un objeto y hacer un seguimiento sobre el mismo, para interpretar su estado. Estos estados pueden ser su forma, como en el caso de utilizar los dedos de la mano para realizar gestos, o simplemente si se reconoce el objeto o no, su posición y movimientos.

3. Diseño

Dentro de este capítulo se va a especificar la manera en la que se afrontará el proyecto para poder cumplir con los objetivos definidos. Primero se escogerá un modelo de desarrollo. Seguidamente se hablará de la planificación que se seguirá teniendo presente el ciclo de vida del software seleccionado. Más adelante se hablará sobre las tecnologías que se usarán para implementar los componentes de nuestro sistema. Por último se especificará la arquitectura del sistema integrando todos sus componentes e indicando como se comunican entre ellos.

3.1. Ciclo de vida del software

Para el desarrollo de este proyecto se ha optado por un modelo de desarrollo incremental. Este modelo se basa en dividir la totalidad del trabajo en partes más pequeñas denominadas incrementos. Estos incrementos o partes se irán desarrollando uno a uno. El primero de ellos es completamente independiente del resto, y los sucesivos, dependientes de los anteriores pues son necesarios para su implementación. Es un modelo modularizado.

Se ha elegido este tipo de desarrollo ya que existen tres secciones o apartados bien separados y distinguibles, que se corresponden con los tres objetivos principales del trabajo. De esta manera, cada uno de los incrementos se corresponderá con uno de los objetivos.

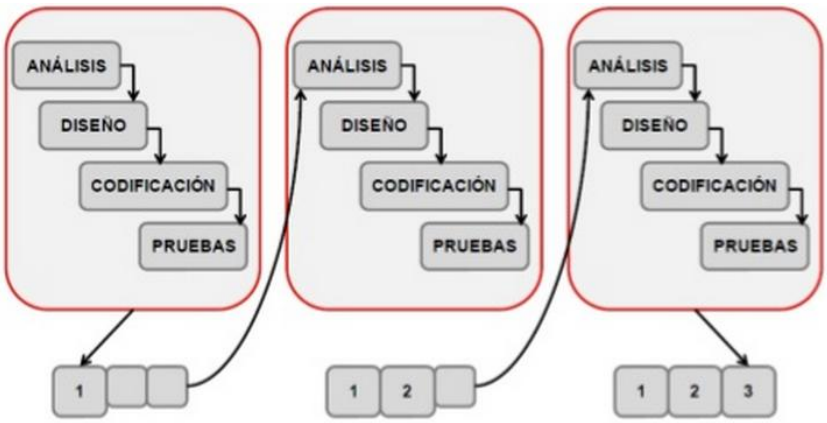

Ilustración VI: Modelo incremental

3.2. Planificación

Siguiendo el modelo de vida del software seleccionado se ha creado una planificación que siguiendo los principios de este modelo permita situar en el tiempo cada una de las tareas necesarias para terminar el proyecto. La planificación realizada puede apreciarse en la Ilustración VII.

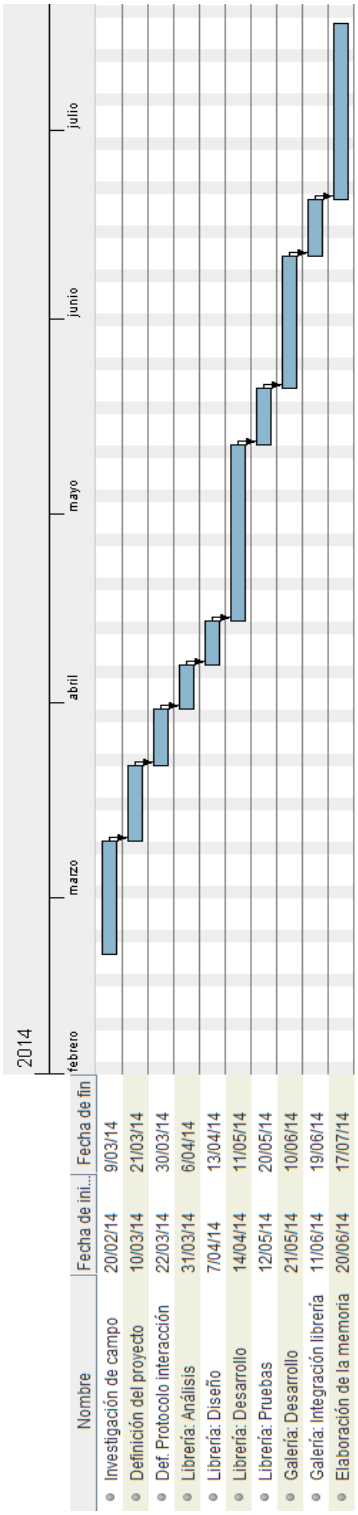

Ilustración VII: Planificación

3.3. Tecnologías a emplear

Dentro de este apartado se va a presentar las tecnologías seleccionadas para la mejor resolución de los objetivos del trabajo. Se hablará de sus características y de por qué son ideales para desarrollar nuestro sistema.

3.3.1. Librería OpenCV

La librería OpenCV (*Open Source Computer Vision*) [15] es libre para su uso tanto con propósitos comerciales como de investigación. Nativamente está escrita en un optimizado C/C++. Está disponible para diferentes lenguajes como Python, Java o C/C++. Existe una reciente versión para Java que, aunque poco documentada, sigue las interfaces de C++.

Se sirve de técnicas de visión artificial para poder procesar imagen o video. Permite así reconocer en imágenes figuras, formas, colores, distancias… y en el caso del video hacer un seguimiento de estas variables para determinar la posición y movimientos de las figuras, las distancias, la variación de colores y prácticamente cualquier cosa que se pueda reconocer en una secuencia de imágenes.

Es una librería muy potente que cuenta con muchas funciones de alto nivel que permiten hacer virguerías en el campo de la visión por computador. Además, aunque se dé el caso de no encontrar implementada una funcionalidad deseada y concreta, la librería dispone de las herramientas a bajo nivel que permitirían implementar la misma.

3.3.2. Java

Se desarrollará tanto la librería de Interacción Natural como la aplicación de la galería fotográfica en Java [16]. De los lenguajes soportados por la librería OpenCV se ha optado por emplear este por varios motivos. En primer lugar las interfaces de la librería son prácticamente iguales a las escritas para C++. También es conocido que las aplicaciones escritas en Java son altamente portables para ser ejecutadas en diferentes máquinas con distintos sistemas operativos.

Además Java cuenta con un paquete llamado Reflection [17] que permitirá a la librería hacer uso de clases externas a la misma. De este modo, cuando la librería detecte uno de los gestos soportados será ella misma la que ejecute el método correspondiente de una clase externa, perteneciente al programa principal que hace uso de la librería. El programa principal tendrá que especificar previamente a la librería de Interacción Natural sobre los objetos y métodos que desee que esta emplee.

3.4. Arquitectura del sistema

De forma general, un sistema que emplee la librería de Interacción Natural objetivo del presente trabajo estará compuesto por las siguientes tres partes, que encajan con cada uno de los elementos a desarrollar para cubrir los correspondientes objetivos del mismo planteados para este TFG.

En primer lugar existirá un programa principal que se tratará de la galería de imágenes. Este programa hará uso de la librería de Interacción Natural desarrollada y así permitirá el reconocimiento por visión por computador de los gestos del protocolo de interacción. Para que esto sea posible el programa principal deberá haber enlazado las funciones correspondientes con cada uno de los gestos disponibles indicándoselo a la librería. A partir de este momento la librería obtendrá las imágenes tomadas por una cámara y empleando las herramientas que proporciona OpenCV junto con la lógica del protocolo de interacción detectará los gestos soportados y podrá ejecutar la función correspondiente. En la Ilustración VIII se puede apreciar un esquema de esta arquitectura.

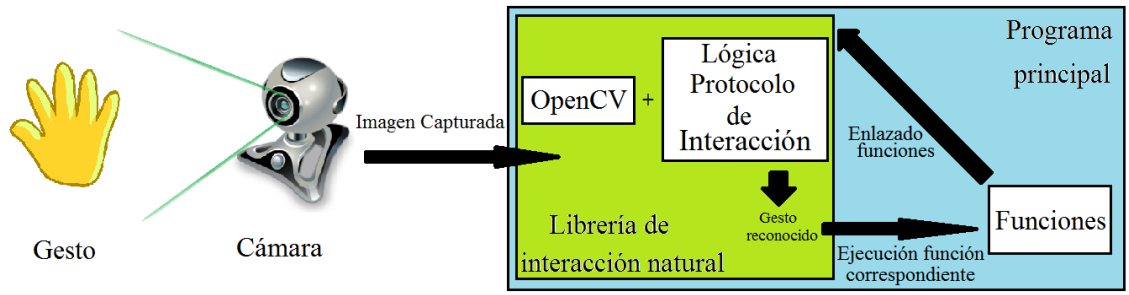

Ilustración VIII: Arquitectura del sistema

4. Desarrollo

En este capítulo se verá cómo se han implementado cada uno de los elementos que componen nuestro sistema. En primer lugar se hablará sobre el protocolo de interacción, planteando cual deberá de ser el conjunto de gestos soportados por este y especificando las acciones necesarias para la detección de los mismos. En segundo lugar se tratará la librería de Interacción Natural viendo las decisiones tomadas para su implementación así de cómo esta se ha llevado a cabo. También se explicará cómo se ha incluido en ella el protocolo de interacción anteriormente presentado para detectar el conjunto de gestos.

4.1. Protocolo de interacción

Se plantea realizar un protocolo de interacción que sea lo más sencillo e intuitivo para el usuario. De esta manera no tendrá que adoptar posturas incómodas o realizar gestos complicados, como sucede en algunos de los proyectos analizados en el estado del arte. Así mismo, la carga computacional para detectar las acciones permitidas por el protocolo será menor al emplear un conjunto reducido y simple. Así no habrá retardos entre la realización de una acción y la respuesta del programa a la misma.

El conjunto de gestos será también reducido y genérico, de tal forma que, con un pequeño número de acciones, se pueda ejecutar prácticamente cualquier acción en un programa que haga uso de este protocolo.

4.1.1 Conjunto de Gestos/Acciones:

Este conjunto constará de tres gestos que denominaremos de **avance**, de **retroceso** y de **selección** respectivamente.

Para la realización e interpretación de los gestos que se van a describir a continuación, serán necesarios dos elementos que interactúen entre sí. Estos podrían ser, por ejemplo, las dos manos del usuario, dos dedos del mismo (uno de cada mano), o dos objetos como dos bolígrafos.

Seguidamente se detalla cada acción soportada y el gesto a realizar según el protocolo. Para poder ilustrarlos se mostraran dos dedos, tal y como se puede comprobar en la Figura IX. Ya que los dedos son distinguibles el uno del otro, en el resto de las ilustraciones se diferenciará a cada uno de ellos mediante una tonalidad distinta.

Ilustración IX: Gesto inicial

1. *Gesto de Avance:*

Para la realización de este gesto se partirá de la posición en que uno de los elementos se encuentra encima del otro como se puede observar en la figura X. Es indiferente cual de los dedos es el que se encuentra encima y cual debajo.

Ilustración X: Gesto avance inicial

Seguidamente será necesario descender el dedo superior hasta quedar bajo el otro, dejando a este último tras él en el proceso. Es decir, habrá que alternar su posición pasando el superior por delante del inferior como se muestra en la figura XI.

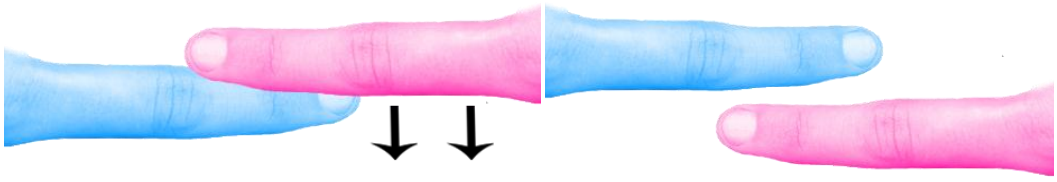

Ilustración XI: Transición 1 del gesto de avance.

Sería equivalente a la acción que se acaba de mostrar el mover el dedo inferior hasta colocarse en la zona superior pero en este caso pasando por detrás del otro dedo. De nuevo se muestra dicha acción mediante la figura XII.

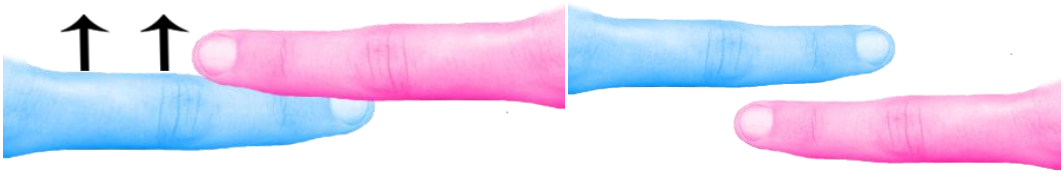

Ilustración XII: : Transición 2 del gesto de avance.

Del mismo modo, sería equivalente combinar ambos gestos. En lugar de desplazar únicamente uno de los dedos, se podrían desplazar ambos en direcciones opuestas, pero

siempre quedando el que partió en la zona inferior por detrás del que se encontraba en la zona superior y viceversa. Esto se muestra en las dos imágenes de la figura XIII.

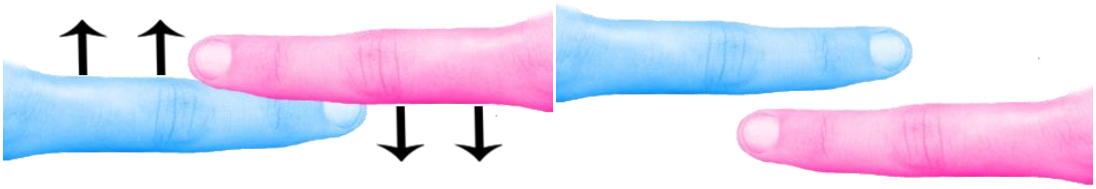

Ilustración XIII: Transición 3 del gesto de avance.

Cualquiera de los movimientos descritos anteriormente, que son todos equivalentes según este protocolo, será interpretado como una acción de avance.

2. *Gesto de retroceso:*

Para realizar la acción de retroceso, se realizará un gesto prácticamente idéntico al de avance, sólo que en esta ocasión será el dedo superior el que quede tras el inferior a la hora de realizar el cambio de posición. Esto queda reflejado en la XIV.

Ilustración XIV: Gesto de retroceso

De nuevo será equivalente mover tan sólo uno de los dos dedos o ambos simultáneamente.

3. *Gesto de selección:*

Para realizar la acción de selección, se partirá de nuevo del estado en que un dedo se encuentra sobre el otro. En este caso, en lugar de cruzarse, deberán de colocarse ambos a la misma altura, en el mismo plano horizontal, tal y como se muestra en las figuras XV y XVI.

Ilustración XV: Transición gesto de selección

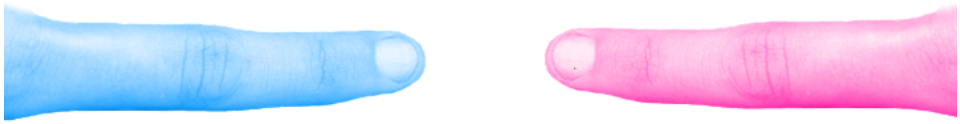

Ilustración XVI: Gesto de selección

4.2. Librería de Interacción Natural

4.2.1. Análisis

La librería de Interacción Natural podría permitir que existiera un gesto o acción que se correspondiese con cada evento de la aplicación que hace uso de ella. Así, en el caso de la aplicación de ejemplo que se empleará (una galería fotográfica), podrían existir diferentes gestos para avanzar a la siguiente foto, retroceder a la anterior, seleccionar una foto, ampliarla, reducir la ampliación, rotar en una dirección, rotar en la otra dirección, etc.

Por el contrario, la Interacción Natural podría simplemente reemplazar las acciones realizadas con los dispositivos de entrada hardware convencionales. De esta manera, aplicándose junto a nuestra aplicación de ejemplo, se necesitarían gestos para avanzar el ratón entre los componentes disponibles, para retrocederlo y para hacer click. Aunque se dispone de un conjunto claramente más reducido que en el caso anterior, y esto podría resultar atractivo, esta interacción no es muy intuitiva.

Uno de los objetivos que se han marcado es que la librería sea altamente portable para su uso en aplicaciones de muy diversa índole. Es por ello que resulta complejo realizar una librería de Interacción Natural donde exista una relación 1 a 1 entre gesto y acción, pues no se sabe cuantas acciones existirán en la aplicación que la use. Pero tampoco podemos limitarnos a mover el puntero del ratón ya que no se obtendría una interacción intuitiva y natural para el usuario. Es por todo lo anterior que se ha tomado la decisión de que la librería a implementar sea mixta. Se permitirá enlazar gestos a acciones concretas dentro del sistema, o que los gestos permitan recorrer un conjunto de componentes disponibles.

También se ha barajado la opción de que la librería desplegase su propia interfaz de usuario que sería complementaria a la de la aplicación que la empleara. Esta opción ha sido descartada ya que sería engorroso superponer una nueva interfaz a la ya existente y no sería

escalable a un gran número de aplicaciones que pudieran tener sus opciones dispersas a lo largo de la aplicación.

4.2.2. Implementación

La librería extenderá a la clase Thread de Java por lo que al emplearla se lanzará un nuevo hilo que se encargará del reconocimiento de los gestos. Para modificar los colores que está tratará de reconocer, existirán unas funciónes (changeColorA y changeColorB) a las que el desarrollador podrá llamar para cambiar los umbrales HSV proporcionando unos nuevos valores máximos y mínimos. La librería dispondrá de dos modos de funcionamiento como se ha mencionado anteriormente. Para cambiar el modo de actuación de la librería será necesario ejecutar las funciones *componentsMode* o *bindingMode*, según se quiera recorrer componentes o enlazar funciones respectivamente.

Modo enlazado de funciones

Para activar este modo se ejecutara la función *bindingMode* como se ha citado anteriormente. Esta función simplemente se encarga de establecer un valor booleano dentro de la librería para reconocer el modo. Seguidamente será necesario indicar qué método de qué objeto habrá que ejecutar tras reconocer cada uno de los gestos soportados en el protocolo de interacción anteriormente definido. Para realizar este emparejamiento entre gesto y método de objeto existen tres funciones dentro de la librería: *BindAdvance*, *BindBack* y *BindSelect*. Estas funciones enlazan el método de un objeto con las acciones avanzar, retroceder y seleccionar respectivamente.

Para que nuestra librería sea capaz de invocar un determinado método de un objeto que a priori no tiene por qué ser capaz de reconocer, se utilizará el paquete *reflect* de java, con el que se puede utilizar reflexión. Esto permitirá a la librería instanciar objetos y métodos en tiempo de ejecución, así como ejecutarlos a partir de sus nombres aunque no se encuentren en su dominio.

Es por ello que las funciones anteriormente mencionadas, *BindAdvance*, *BindBack* y *BindSelect*, recibirán el objeto sobre el que habrá que ejecutar un método al detectar el gesto correspondiente junto con el nombre de la clase a la que pertenece el objeto y el nombre de su método. Las funciones obtendrán la clase a partir del nombre proporcionado gracias a la función *Class.forName*. Después obtendrán el método al ejecutar la función *getMethod* de *reflect* sobre la clase obtenida.

Lo último que hay que entender sobre este modo, es que cuando se detecte un gesto de avance, retroceso o selección del protocolo de interacción anteriormente explicado, se ejecutará la función *invoke* de *reflection* sobre el método obtenido correspondiente a dicho gesto, pasando como argumento el objeto sobre el que invocar dicho método.

Modo recorrido de componentes

Para activar este modo se ejecutara la función *componentsMode* como se ha citado anteriormente. Esta función simplemente se encarga de establecer un valor booleano dentro de la librería para reconocer el modo.

Seguidamente habrá que proporcionar una colección de componentes junto con la ventana en la que se encuentran. Esto se hará mediante la función *BindComponents(JFrame f, Vector<JComponent> cs)*. Esta función guardará la posición de cada uno de los componentes dentro de su ventana en una colección de puntos así como la propia ventana donde se encuentran para poder calcular la posición final de los componentes. También guarda el número de componentes que existen e inicializa un contador a 0 para indicar que se está seleccionando el primero de los componentes. Por último llama a la función *moveMouse* la cual se encarga de desplazar el puntero del ratón a la posición final del componente señalado por el contador, en este caso el primero de ellos.

Lo último que hay que entender sobre este modo, es que cuando se detecte un gesto de avance del protocolo de interacción anteriormente explicado, se ejecutará la función *NextButton*. Esta función incrementará el contador que indica el componente actual y ejecutará la función *moveMouse* para situar el ratón en el lugar correspondiente. Algo similar ocurrirá al detectar un gesto de retroceso: se ejecutará la función *PrevButton*, que decrementará el valor del contador que indica el componente actual y se llamará del mismo modo a *moveMouse* para situar el ratón sobre este. En el caso de detectar el gesto de selección se llamará a la función *SelectButton*, que no hace otra cosa que lanzar el evento de pulsación del botón izquierdo del ratón. Para poder ejecutar acciones en nombre del ratón se empleará la clase Robot de Java.

Diagrama de secuencia UML

Para dejar más claro el modo en el que se ha implementado la librería y cómo funciona, se incluye en la Ilustración XVII un diagrama de secuencia de una aplicación (Main) que ejecuta la librería de Interacción Natural (NaturalInteraction) en el modo de enlazado de funciones y posteriormente en el de recorrido de componentes. En el primero de ellos se enlazan los métodos de un objeto llamado “obj” de la clase ObjectX. En el segundo se emplea la clase Robot de Java como se ha mencionado para controlar el cursor del ratón.

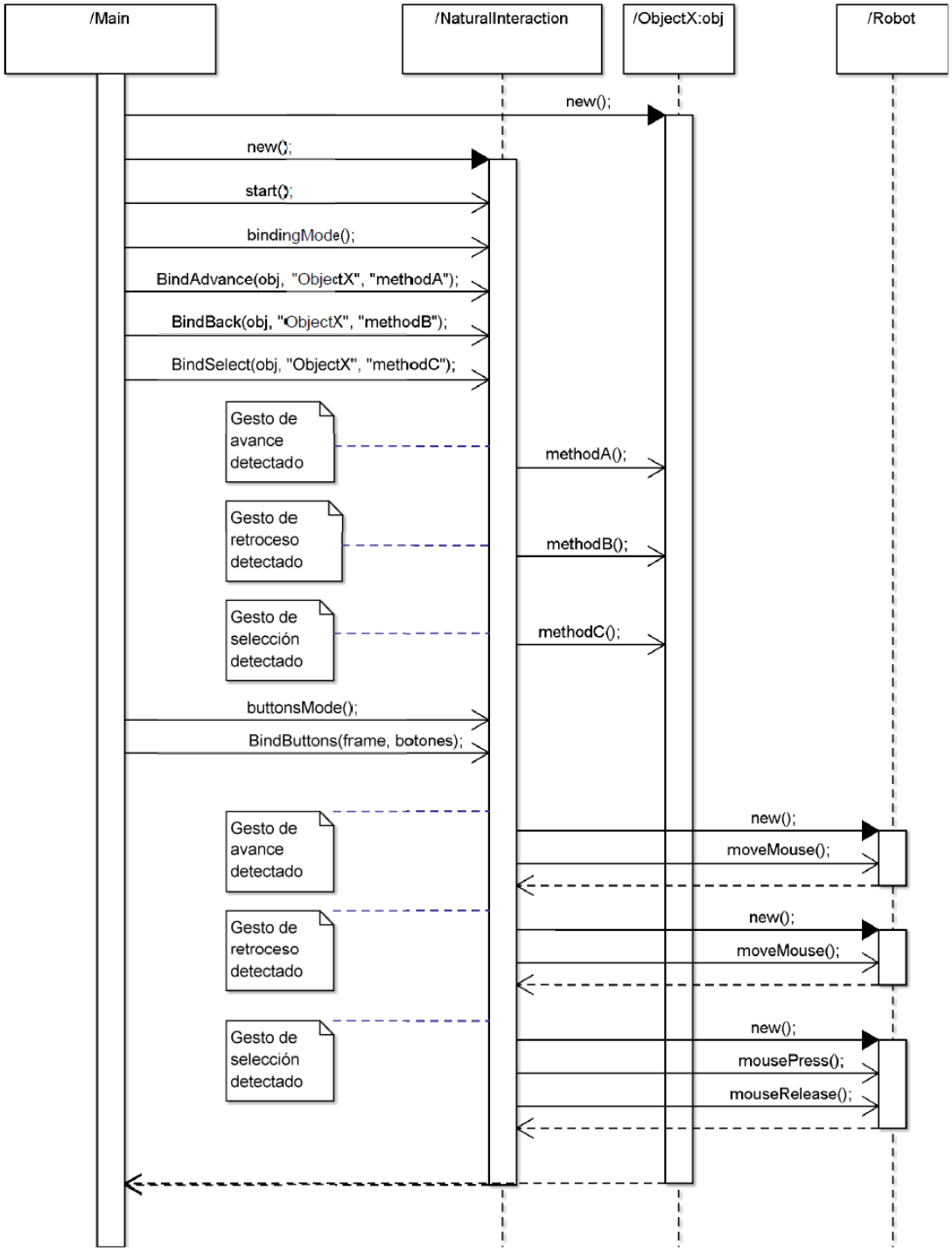

Ilustración XVII: Diagrama de secuencia UML

4.2.3. Identificación, seguimiento y detección.

Para la identificación de los dos elementos que intervienen en el protocolo de interacción anteriormente diseñado, se ha decidido emplear la técnica de detección de colores. Seguidamente será necesario hacer un seguimiento de dichos elementos para poder determinar qué gesto del protocolo están representando y realizar la acción correspondiente al gesto y el modo en el que se encuentre la librería.

El uso de una técnica de detección de colores implica que el color de cada uno de los dos elementos a emplear en el protocolo deberá de ser distinto del otro. Del mismo modo, no deberá haber ningún otro objeto en el campo de visión de la cámara con el mismo color que alguno de los dos elementos, pues resultaría como una interferencia. Aún no habiendo ningún elemento con el mismo color que los elementos citados, es posible que reflejos o cambios en la luz del entorno produzcan estas interferencias, por lo que se han tenido ciertas precauciones que permitirán pasar por alto las mismas.

A continuación se detallan todos los procesos que intervienen a la hora de que la librería sea capaz de realizar estas tres acciones.

1. Proceso para la identificación de uno de los elementos:

En primer lugar se obtiene una imagen capturada por la cámara web, primera imagen de la figura XVIII. Esta imagen se convierte de formato RGB a formato HSV (*Hue, Saturation, Value* – Matiz, Saturación Valor) (segunda imagen de la figura XVIII), que permitirá distinguir con mayor facilidad el color buscado.

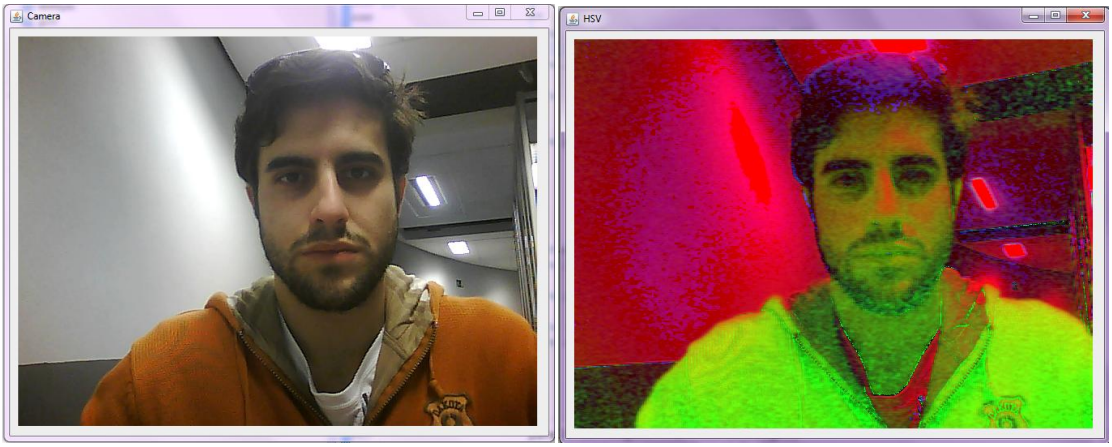

Ilustración XVIII: Conversión a formato HSV

Seguidamente se aplica un filtro, indicando los valores HSV mínimos y máximos entre los que se espera encontrar nuestro elemento. De esta manera se obtiene una imagen en binario, que nos indica si cada pixel de la imagen se encuentra o no entre los valores del filtro. Esta imagen se obtiene gracias a la función *inRange* de OpenCV que permite aplicar el filtro de color mencionado.

Durante este proceso también es necesario aplicar un desenfoque gaussiano o una suavización gaussiana. Esto nos permitirá eliminar pequeñas interferencias que hayan sido detectadas al aplicar el filtro de color pero no se correspondan con nuestro objeto. Para tal efecto se ha aplicado la función *GaussianBlur* de OpenCV.

En la figura XIX se puede observar el proceso completo para la identificación del tapón de un rotulador azul.

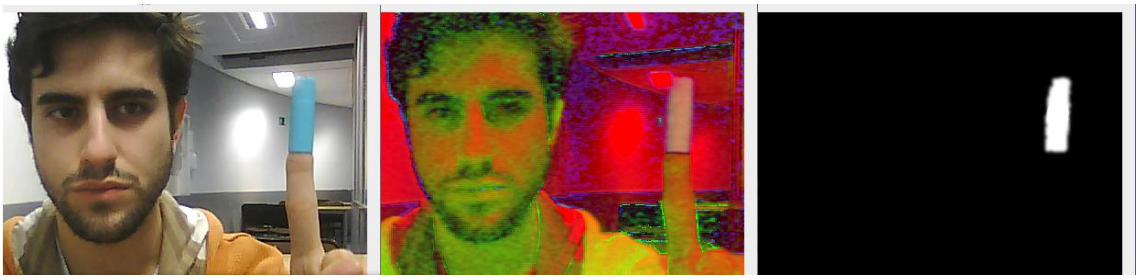

Ilustración XIX: Obtención de imagen en binario

En el siguiente cuadro se presenta un pseudocódigo del proceso anteriormente descrito a modo de resumen:

```
capture.read(webcam_image); //Capturar imagen de la cámara

Imgproc.cvtColor(webcam_image, hsv_image, BGR_TO_HSV) //Convertir imagen de RGB a HSV

Core.inRange(hsv_image, hsv_min, hsv_max); //Filtro de la imagen según el rango de colores

Imgproc.GaussianBlur(hsv_image); //Suavizado de la imagen
```

2. *Proceso para el seguimiento de uno de los elementos:*

Lo siguiente necesario para poder detectar nuestros gestos es tener noción de donde se encuentran nuestros elementos. Esto es posible al buscar contornos en nuestra imagen binaria, obtenida tras aplicar el filtro de color y el desenfoque gaussiano. De esta manera si hubiera varios objetos en blanco en la anterior imagen binaria se obtendrían los contornos de ambos objetos. Esta acción se ha llevado a cabo empleando la función *findCountours* de OpenCV, que nos devuelve todos los contornos detectados.

Se puede obtener el área de cada uno de estos contornos obtenidos gracias a la función *CountourArea* de nuevo de OpenCV. Así nos quedaremos tan solo con uno de los varios posibles contornos obtenidos. El criterio será el de guardar la información referente al contorno de mayor área. Se supone que las interferencias deberán de ser de menor tamaño que el objeto a identificar, pues de otro modo sería imposible detectar el objeto al existir demasiada interferencia en la imagen. Para explicar esto más claramente, imaginemos que el usuario lleva un anillo azul. Sería sencillo eliminar el contorno del anillo que sería relativamente pequeño. En cambio, si existiera una interferencia de gran tamaño como que la camiseta del usuario fuera azul, o la pared del lugar donde se encuentra fuera de este color, sería inviable detectar el objeto.

Tras elegir el contorno en el que se está interesado solo se necesitará obtener el centro de masa del mismo empleando la clase *Moments* de OpenCV.En la figura XX se muestra como tras encontrar el centro de masa del objeto se ha pintado un círculo en el punto correspondiente.

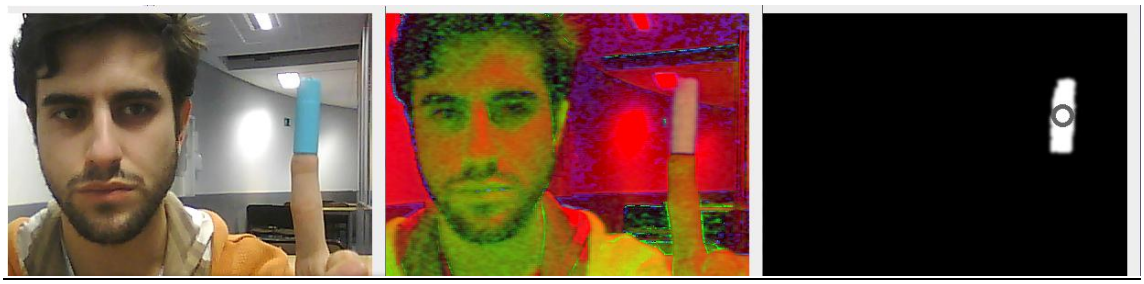

Ilustración XX: Detección de la posición de un elemento

Para llevar un seguimiento de la trayectoria y posición del objeto simplemente es necesario guardar el centro de masa del mismo para que esté disponible para ser consultado en el reconocimiento del siguiente fotograma. Seguidamente se muestra un cuadro con el pseudocódigo.

```
Imgproc.findContours(hsv_image, contoursList); //Obtener contornos
maxArea=0;

recorrer (contoursList.size()) { //Recorrer contornos
        area = Imgproc.contourArea(contoursList.get(i)); //Obtener área de un contorno
        si  (area > maxArea) {
                maxArea = area;          //Seleccionar el área con mayor contorno
                selectedCountour = i;
        }
}

moments = Imgproc.moments(contoursList.get(selectedCountour)); //obtener momentos del
                                                                //contorno seleccionado
mc = Point(moments.get_m10() / moments.get_m00(), moments.get_m01() /
moments.get_m00()); //Obtener centro de masa a partir de los momentos del contorno
```

3. *Proceso para la detección de gestos del protocolo de interacción:*

Los procesos de identificación y detección de objetos anteriormente citados se aplicarán para dos colores. Esto permitirá disponer de dos objetos que interactúen entre sí tal y como se espera según el protocolo de interacción diseñado.

Como se ha avanzado en la sección anterior se empleará el centro de masa de cada objeto en el fotograma anterior para saber qué es lo que está ocurriendo. De esta manera, si el punto correspondiente al centro de masa de uno de los objetos del fotograma anterior tiene un valor mayor en la coordenada Y que el actual centro de masa para este mismo objeto, se puede deducir claramente que este se ha desplazado hacia arriba. Exactamente de la misma manera, si la coordenada Y del anterior centro de masa fuera de menor valor que la actual, implicaría que el objeto se ha desplazado hacia abajo.

La detección de los gestos de avance y retroceso serán muy similares por sus grandes similitudes e implicaciones entre ambos. A continuación se detallará la lógica para la detección de estos gestos:

- En primer lugar, se tendrá en cuenta que los objetos se encuentren uno encima del otro. Esto se apreciará al comprobar que la diferencia entre las coordenadas X de sus centros de masas se encuentra dentro de un cierto umbral.

-A continuación se prestará atención a las coordenadas Y de los centros de masa de ambos objetos. Esto nos permitirá apreciar cuál se encuentra encima de cuál (o debajo) y se guardará esta información para que se encuentre disponible en la detección de los objetos del siguiente fotograma.

- Seguidamente se vuelve a centrar la atención en las coordenadas Y de los centros de masa y así se sabrá si se encuentran muy próximos el uno con respecto al otro. Para ello se obtiene la diferencia entre las coordenadas de ambos. Si este valor es menor que un cierto umbral implicará que ambos objetos están verdaderamente cerca.

- Tras detectar que los objetos están muy cerca en el eje Y, se sabrá que los objetos están próximos a cambiar de posición, de estar uno encima con respecto al otro a pasar a estar debajo. Será entonces cuando se compararán las áreas de los contornos de estos objetos, para saber cual tiene menor área. El objeto con menor área estará situado detrás del otro objeto, ya que se verá eclipsado por el mismo. El objeto que se encuentra por delante conservará todo su área visible y esta será por lo tanto mayor que la del objeto que está ocultando. De nuevo se guardará la información referente a cuál de los dos objetos se encuentra detrás para que esté disponible en la detección de los objetos del siguiente fotograma.

-Finalmente solo queda detectar un cambio en la posición entre los objetos, que el que se encontraba encima ha pasado a estar debajo o viceversa. Esto será posible comparando la información relevante a la posición de los objetos del anterior fotograma con la actual. Una vez esto haya sido detectado habrá cuatro posibles escenarios:

1. El objeto 1 se encontraba encima y ha pasado debajo por delante del objeto 2.
2. El objeto 1 se encontraba encima y ha pasado debajo por detrás del objeto 2.
3. El objeto 2 se encontraba encima y ha pasado debajo por delante del objeto 1.
4. El objeto 2 se encontraba encima y ha pasado debajo por detrás del objeto 1.

Los casos 1 y 3 serían análogos, tan solo variaría que objeto era el que se encontraba encima o debajo, pero ambos indicarían un gesto de avance. Lo mismo ocurriría para el par de casos 2 y 4 pero en este caso se habría detectado un movimiento de retroceso.

La detección del gesto de selección no es tan compleja como la de avance y retroceso. En este caso tan solo se necesita detectar que al contrario que en los casos anteriores no se encuentran uno encima del otro, si no que se encuentran al lado. Esto se logrará obteniendo la diferencia entre las coordenadas X de ambos objetos, la cual ha de ser mayor a un cierto umbral para considerar que efectivamente se ha realizado un gesto de selección colocando un objeto al lado del otro.

Lo que sucederá tras la detección de alguno de los gestos dependerá como se ha visto anteriormente del modo en el que se encuentre la librería, en el de recorrer componentes o el de enlazar funciones.

5. Evaluación de la librería de Interacción Natural: Caso de estudio con una aplicación de ejemplo (Galería de imágenes)

Llegados a este capítulo se probarán y mostraran las funcionalidades implementadas en nuestra librería de Interacción Natural a través de una aplicación de ejemplo, una galería de imágenes. En primer lugar se presentará la aplicación galería de imágenes para así familiarizarnos con ella y conocer las funcionalidades de las que dispone. Seguidamente se mostrará como se ha implantado la librería desarrollada a la galería de imágenes. Se hablará de las posibilidades existentes para el uso de la librería en distintos escenarios.

5.1. Funcionalidad de la aplicación de prueba

La aplicación creada como ejemplo para mostrar el uso de la librería de Interacción Natural se trata de una galería de imágenes tal como se ha adelantado. Seguidamente se verá que es capaz de hacer esta aplicación mostrando sus distintas funcionalidades.

Esta galería nos permite visualizar y realizar distintas operaciones sobre una serie de imágenes situadas en una carpeta. Más concretamente, la carpeta en cuestión se encuentra en la ruta C://migaleria. Al ejecutar la aplicación se nos muestra la primera de las imágenes junto con una serie de botones que nos permiten retroceder a la anterior, avanzar a la siguiente o seleccionar la imagen que estamos viendo para realizar alguna operación sobre la misma. En la barra superior de la ventana se muestra el nombre del archivo. En la figura XXI se muestra la pantalla de la aplicación una vez se ha cargado.

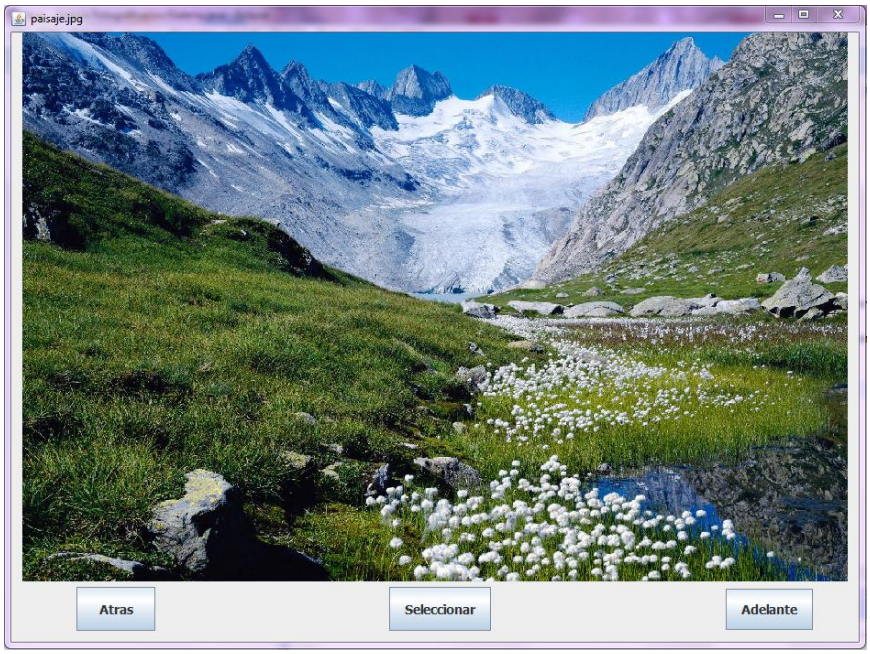

Ilustración XXI: Exploración de imágenes

Al seleccionar una imagen, los botones inferiores desaparecen dando lugar a una nueva serie de opciones, en esta ocasión relacionados con las operaciones que se permite realizar a la fotografía para su manipulación. Esta nueva pantalla se corresponde con la figura XXII.

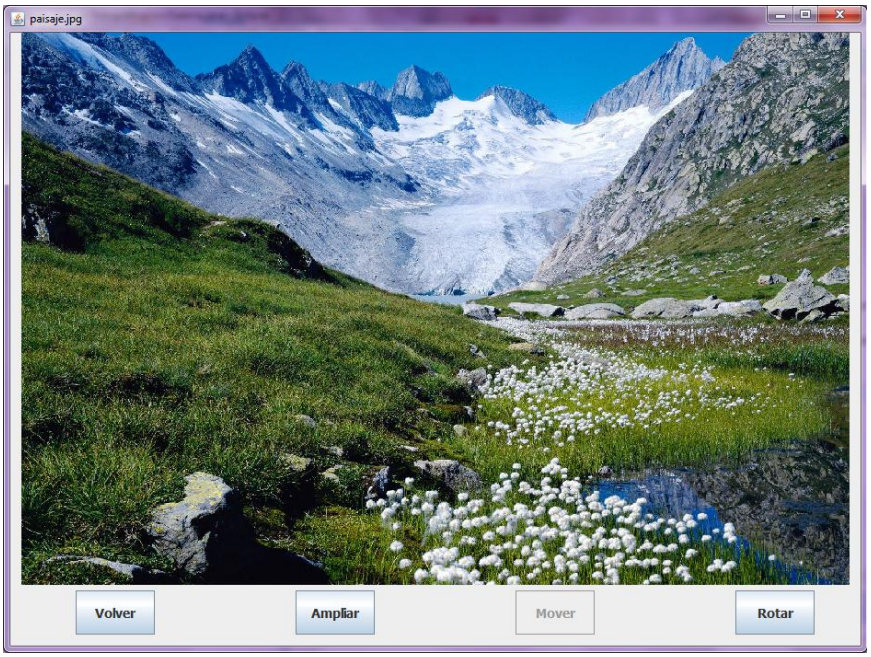

Ilustración XXII: Funcionalidades de imagen

El botón volver nos permite retroceder para elegir una nueva fotografía de entre las disponibles. Mediante el botón ampliar se nos da la opción de agrandar la imagen para verla en más detalle.

Al seleccionar la opción de ampliar se nos muestra una nueva pantalla con tres botones, uno de ellos para ampliar más, otro para reducir la imagen y uno más para terminar la operación. Esta pantalla aparece reflejada en la figura XXIII.

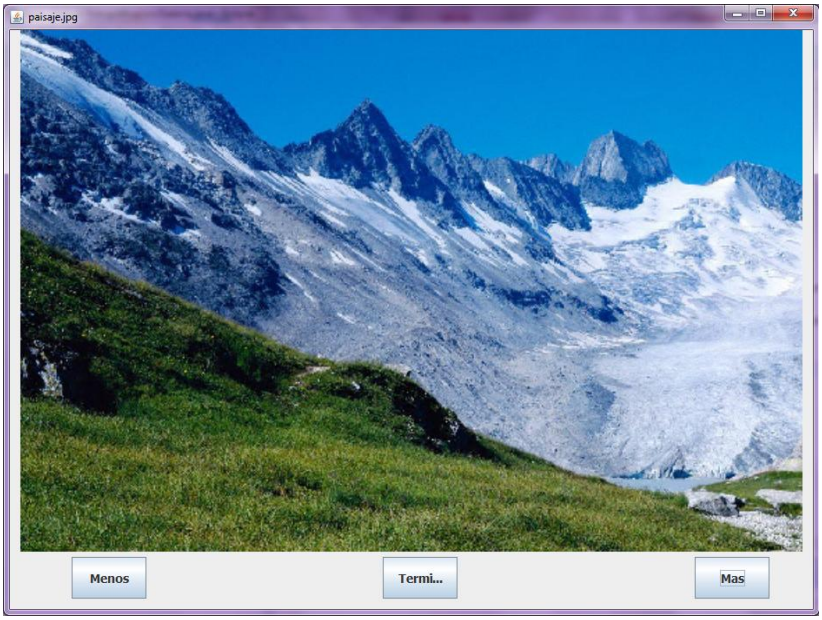

Ilustración XXIII: Funcionalidad de ampliación

La siguiente opción disponible es la de Mover. Esta opción está habilitada cuando la imagen está ampliada y no se muestra por completo. Permite desplazarse a lo largo de la

imagen hacia arriba, abajo, derecha e izquierda. Esta nueva ventana se puede observar en la figura XXIV.

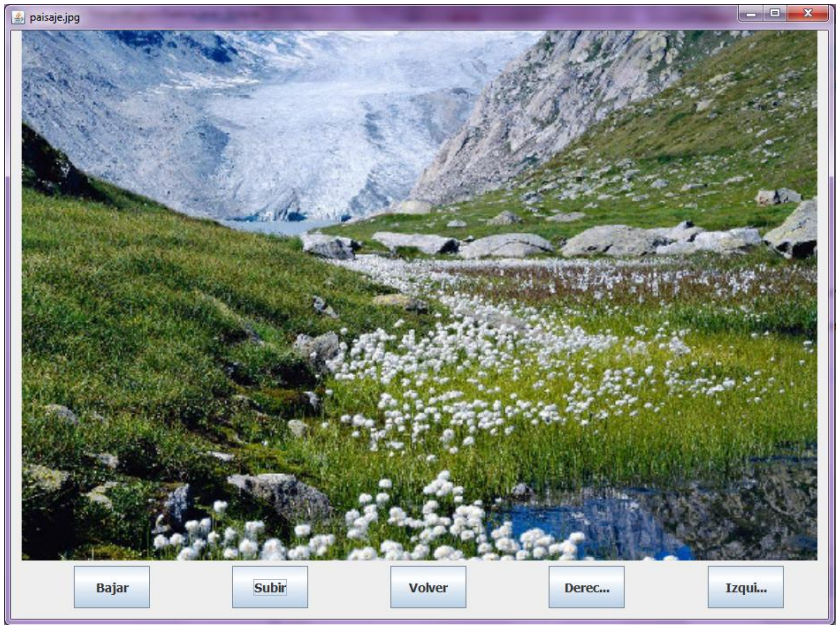

Ilustración XXIV: Funcionalidad mover

Por último, mediante el botón rotar se nos permite realizar una rotación sobre la imagen, en el sentido de las agujas del reloj o al contrario. La operación puede ser descartada o confirmada. En este último caso la operación se hace persistente sobre el archivo original. La nueva pantalla que aparece tras pulsar el botón rotar y que permite realizar estas operaciones es la correspondiente con la figura XXV.

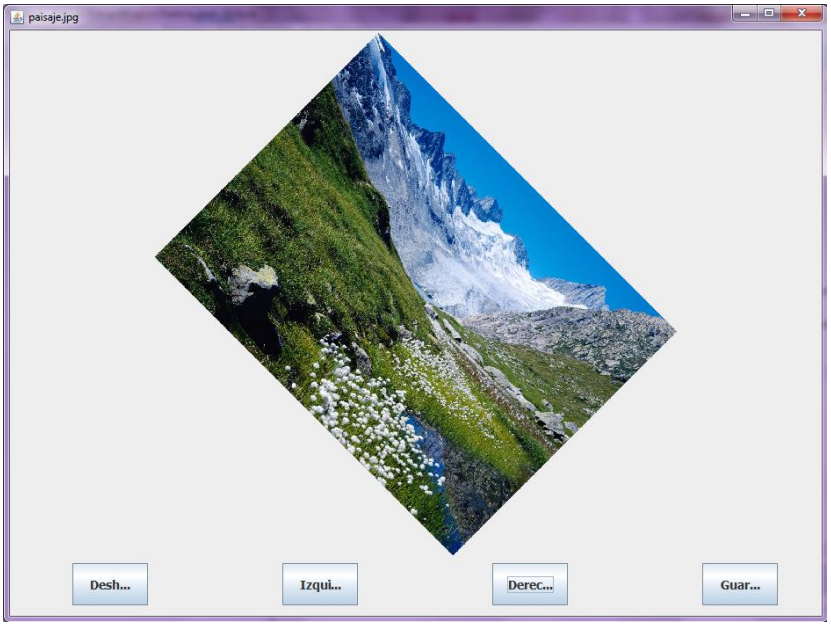

Ilustración XXV: Funcionalidad rotar

5.2. Empleo de la librería de Interacción Natural en la aplicación de prueba

Para que la aplicación de la galería pueda beneficiarse de la Interacción Natural que permite la librería desarrollada será necesario que esta la controle a su antojo, en primer lugar ha de instanciar un objeto de la clase *NaturalInteraction*, seleccionar el modo en el que va a trabajar y llamar a los métodos de inicialización para dicho modo. Por último ha de invocarse el método *start* para arrancar el hilo que se encarga de detectar los gestos.

En el caso concreto de esta aplicación, se comienza con el modo de enlazado de funciones llamando al método *bindingMode* como se explicó con anterioridad. Así, seguidamente enlaza el gesto avanzar con el método *fotoAdelante* de la clase Galería, el gesto retroceder con el método *fotoAtras* y el gesto seleccionar con el método *fotoSeleccionar*. Ahora se lanzará el hilo de reconocimiento mediante el método *start*.

De aquí en adelante, tras la inicialización, se deberá ir cambiando el modo de actuación de la librería así como los métodos a ejecutar según sea conveniente. De esta manera se genera una máquina de estados que permitirá la transición entre los mismos según se recorran las distintas opciones del programa. De esta manera, según la lógica de la aplicación, al llamar al método *fotoSeleccionar* habrá que volver a reajustar la librería. En este caso se ha optado por utilizar el otro modo existente, con el que recorrer una serie de componentes. Es por ello que se cambia el modo mediante el método *componentsMode* como ya se ha visto. Para terminar se debe proporcionar la lista de componentes implicados en este modo para que se puedan explorar.

Si ahora se selecciona el botón volver, sería como volver al estado anterior en esta máquina de estados, y sería necesario volver a realizar las mismas llamadas que en la inicialización a excepción de la llamada al método *start*. Para tener una visión más gráfica de esta máquina de estados que se crea se puede observar la Ilustración XXVI donde queda reflejada.

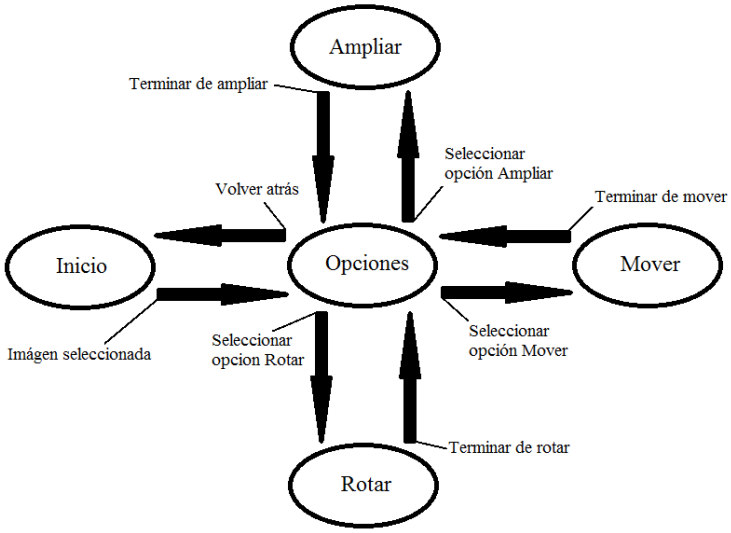

Ilustración XXVI: Diagrama de estados

La opción de ampliar vuelve a emplear el modo de enlazado de funciones. Más concretamente se enlazan el gesto avanzar con ampliar más, el de retroceder con reducir la ampliación y el de seleccionar con el de terminar la operación. Al existir tres funciones lo más recomendable es emplear este modo, así como porque se produce una relación natural entre las operaciones y los movimientos gestuales reconocidos por la librería. Aún así el desarrollador de la galería que incluye la librería podría haber empleado el modo de recorrer componentes, aunque obteniendo una experiencia mucho menos agradable para el usuario.

La opción mover cuenta con 5 posibles opciones como se ha visto. Estas son las de desplazarse arriba, abajo, a derecha o izquierda a lo largo de una foto ampliada y la opción para terminar. El modo empleado por la librería vuelve a ser el de enlazado de funciones. Se ha enlazado el gesto avanzar con el método que permite desplazarse hacia la derecha, el gesto de retroceso con el método para subir y el gesto de selección con la opción de volver. No todas las opciones son accesibles empleando los gestos de la librería natural pero se permite el desplazamiento a lo largo de toda la foto aún sin poder desplazarse hacia abajo o a la izquierda pues al llegar al final de la foto se vuelve al extremo opuesto. Se podría permitir al usuario alcanzar todas las opciones habilitando el modo de recorrer los componentes, aunque este ralentizaría la manera en la que el usuario interactúa con la aplicación así como reduciría la naturalidad de dicha interacción.

Si la opción seleccionada al recorrer los botones es la de rotar, se vuelve al modo de enlazado de funciones. De manera similar a recorrer las imágenes, se permite rotar hacia un lado u otro con los gestos de avanzar y retroceder. El gesto de selección se corresponderá con la función guardar. Es este caso nos queda una posible opción no alcanzable con la librería que se corresponde con la de descartar la rotación realizada. Esto se podría solucionar empleando el modo de recorrer componentes, pero se ha considerado más adecuado que no se alcance pues es una manera más intuitiva y rápida de uso y se puede descartar una rotación realizando otra o dejándola en la posición inicial. Además, si el programa detecta que el ángulo de rotación a aplicar es de cero grados la acción detectada al guardar es la de descarte. El modo a emplear la librería quedaría completamente a decisión del desarrollador que la incluya en su programa, según él considere oportuno.

Para que le hilo que se encarga de reconocer los gestos finalice se incluye la llamada al método *Stop* de la librería cuando se detecta el cierre de la galería.

Las pruebas realizadas mediante el empleo de la librería en la aplicación revelaron que esta cumplía con las expectativas iniciales. Permitía una interacción ágil con el programa sin incluir retrasos por la detección de los gestos.

6. Conclusiones

En este capítulo se expondrán los resultados obtenidos tras la elaboración del proyecto. Se verá en qué medida se han alcanzado los objetivos planteados, se identificarán posibles mejoras y trabajos futuros, y se valorará de manera personal la experiencia al desarrollarlo.

6.1. Consecución de objetivos

Valorando el producto final que se ha desarrollado, se puede concluir que el trabajo llevado a cabo ha conseguido cumplir con todos los objetivos planteados.

Así, para el objetivo 1, el protocolo de interacción diseñado está compuesto por unos gestos simples e intuitivos. Tan solo se han especificado tres gestos, lo que implica que el aprendizaje de su uso será muy rápido y sencillo. Además estos gestos son ambidiestros, a pesar de emplear ambas manos para su ejecución no existe distinción entre la derecha y la izquierda.

Por otro lado, para el objetivo 2, la librería soporta dos modos de funcionamiento, uno que enlazará los gestos con las acciones disponibles en la aplicación y otro que mediante estos gestos recorrerá los componentes de la misma para poder hacer click sobre ellos y lanzar las acciones vinculadas a los mismos. Esto permitirá a los desarrolladores que la empleen seleccionar el modo más conveniente para cada escenario de sus aplicaciones. En el caso de que en alguno de estos escenarios haya un número de actuaciones posibles mayor que el número disponibles de gestos, haciendo imposible una correspondencia 1 a 1 entre acción y gesto, se podrán emplear los gestos existentes para desplazarse por los componentes que ejecutan estas acciones para acceder a las mismas. Todo esto hace que la librería sea altamente portable para cualquier tipo de aplicación que se valga de componentes Java para su uso. Así mismo, será posible su uso en diversas máquinas que se valgan de diferentes sistemas operativos ya que Java se encuentra ampliamente extendido.

Por último, para el objetivo 3, tras las pruebas realizadas a esta librería al hacer uso de ella junto con la aplicación de prueba, la galería de imágenes, se ha demostrado que esta es capaz de identificar los tres gestos detectados y de lanzar la acción pertinente según se haya especificado. Además se han demostrado las posibilidades que proporciona al disponer de los dos modos anteriormente citados.

Cubiertos todos los objetivos, puede verse cómo se ha trabajado sobre la motivación principal del presente TFG, aportando alternativas a las clásicas interacciones del usuario con el computador, utilizando gestos realizados por sus propias manos para indicarle al sistema qué es lo que desea hacer.

6.2. Posibles mejoras y trabajos futuros

A pesar de que se ha conseguido cumplir con los objetivos contraídos al inicio del trabajo, durante el desarrollo del mismo se han observado posibles mejoras que aumentarían la facilidad en el uso de la librería así como su integración con las aplicaciones que la empleen.

En primer lugar, la librería desarrollada trabaja detectando dos colores predefinidos para la detección de los gestos. Tan solo se soporta la variación de estos valores ejecutando una función de la librería por parte del desarrollador para modificarlos. Sería muy aconsejable que esto no fuera así ya que es posible que el usuario se encuentre en un entorno con un fondo que incluya dichos colores predefinidos y le imposibilitaría el correcto funcionamiento de la librería. La variación de estos valores podría hacerse introduciendo los límites máximos y mínimos de los colores en HSV esperados en texto al iniciar la aplicación. De una manera más sofisticada se podría mostrar al usuario una ventana que enseñase en tiempo real lo recogido con la cámara web para situar sobre esta el objeto que se desea identificar y fuese la propia librería la que estipulase los valores de color en HSV más adecuados para dicho objeto.

Otra posible mejora consistiría en facilitar aún más la integración de la librería con las aplicaciones que hagan uso de ella. Para ello, en lugar de ser necesario llamar a las funciones de la librería que la controlan, habría que especificar los estados que existen en la aplicación y cómo se desea que actúe la librería en cada uno de ellos. De esta manera se podría concretar una máquina de estados donde una determinada acción en un estado hace que se pase a otro de los estados disponibles. Esto sería posible por ejemplo mediante un archivo XML que contuviera toda esta información, los estados disponibles, incluyendo el estado inicial, las acciones de las que disponen y las transiciones de estado de cada acción en caso de haberlas. Al iniciar la aplicación habría que pasar este archivo a la librería para que pudiese comportarse según lo esperado, siendo ella la encargada de ejecutar los métodos que correspondan.

Por último, sería interesante contar con un conjunto algo mayor de gestos, que aunque complicarían algo el uso final del usuario, posibilitarían a aplicaciones que cuenten con muchas acciones disponer de uno de los gestos para cada una de ellas, minimizando así el uso del modo de recorrido de componentes que es menos intuitivo. Cada aplicación podría elegir qué conjunto de gestos emplear de entre los disponibles por la librería.

6.3. Valoración personal

El campo de la Interacción Natural es algo cada vez más presente en nuestras vidas. Sin ir más lejos estamos rodeados de teléfonos inteligentes que cuentan con pantallas táctiles que nos permiten interactuar con ellos al tocarlos. En el pasado era una idea casi futurista, pero hoy en día, como se ha visto en el estado del arte, existen numerosas tecnologías que permiten comunicarnos con los sistemas de maneras que antes eran impensables.

Antes de comenzar a desarrollar el proyecto tenía la impresión de que sería complicado de abarcar, pues hacer que los sistemas informáticos interpreten el mundo que los rodea no parece simple a primera vista. De hecho no es una tarea sencilla, pero gracias a nuevas tecnologías y herramientas, como la Visión por Computador y la librería OpenCV que implementa multitud de algoritmos, es posible simplificar esta tarea. Realizar este trabajo me

ha permitido ser consciente de las grandes posibilidades que existen en el ámbito de la Interacción Natural, investigando los proyectos más modernos y dejando volar la imaginación. A pesar de llevar a cabo un proyecto modesto, se han obtenido resultados muy gratificantes y eso ratifica las grandes posibilidades de los avances que existen hoy en día. Ahora solo me cabe esperar ver hasta dónde puede llegar la Interacción Natural entre hombres y máquinas y posiblemente aportar nuevas ideas y tecnologías para poner mi granito de arena en este campo.

7. Referencias

[1]. Leap Motion https://www.leapmotion.com/ consultado en junio del 2014.

[2]. Structure http://structure.io/ consultado en junio del 2014.

[3]. Kinect http://www.xbox.com/es-ES/Kinect consultado en junio del 2014.

[4]. PlayStation Move http://es.playstation.com/psmove/ consultado en junio del 2014.

[5]. Amazon Fire Phone http://www.amazon.com/Fire_Phone_13MP-Camera_32GB/dp/B00EOE0WKQ consultado en junio del 2014.

[6]. William T. Freeman, P. A Beardsley, H. Kage, K. Kyuma, C. D. Weissman: "*Computer vision for computer interaction*", SIGGRAPH Computer Graphics magazine http://www.merl.com/publications/docs/TR99-36.pdf consultado en junio del 2014.

[7]. G. Bradski y A. Kaehler (2008): "*Learning OpenCV: Computer Vision with the OpenCV Library",* O'Reailly Media

[8]. F. Jurado, C. González (2014): "*Interfaces de Usuario Avanzadas*" capitulo 31 en "*Desarrollo de Videojuegos, Un Enfoque Práctico*" http://www.cedv.es/descargas/cedv_3Ed.pdf consultado en junio del 2014.

[9]. X. Zabulis, H. Baltzakis, A. Argyros (2009): "*Vision-based Hand Gesture Recognition for Human-Computer Interaction*" Capítulo 34 en "The Universal Access Handbook", Lawrence Erlbaum Associates, Inc. (LEA), Series on "Human Factors and Ergonomics" http://www.ics.forth.gr/_publications/2009_06_book_hci_gestures.pdf consultado en junio del 2014.

[10]. T. M. Alisi, G. D'Amico, A. Del Bimbo, A. Ferracani, L. Landucci, N. Torpei (2009): "*Natural Interaction at Work*" http://tom.londondroids.com/wp-content/uploads/2009/04/eva09.pdf consultado en junio del 2014.

[11]. Paul Goh (2005): "*Automatic Recognition of Auslan Finger-spelling using Hidden Markov Models*", Tesis doctoral, University of Westem Australia http://undergraduate.csse.uwa.edu.au/year4/Current/Students/Files/2005/PaulGoh/CorrectedDissertation.pdf consultado en junio del 2014.

[12]. Suad Seirafy, Fatima Zubaidi (2012): "*Hand Gestures Recognition Application "Virtual Mouse" "Viertual Piano" "Integration with Interactive Game""*, Informe de Proyecto Final de Carrera, An-Najah National University Faculty of Engineering http://eng.najah.edu/sites/eng.najah.edu/files/software_graduation_project_documentation.docx consultado en junio del 2014.

[13]. Aamod Kore, Nisheeth Lahoti, S. S. Kausik, Kshitij Singh Mehlawat (2012): "*Live Gesture Recognition*" http://www.cse.iitb.ac.in/~aamod/glive/itsp.html consultado en junio del 2014.

[14]. Oleksiy Busaryev, John Doolittle: "Gesture *Recognition with Applications*", informe técnico http://web.cse.ohio-state.edu/~busaryev/Projects/Gesture%20Recognition%20with%20Applications/Report.pdf consultado en junio del 2014.

[15]. OpenCV http://opencv.org/ consultado en junio del 2014.

[16]. Java https://www.java.com/es/ consultado en junio del 2014.

[17]. Java Reflection http://docs.oracle.com/javase/tutorial/reflect/ consultado en junio del 2014.

[18]. Tracking a ball with Java/OpenCV http://cell0907.blogspot.com.es/2013/07/tracking-ball-with-javaopencv.html consultado en junio del 2014.

[19]. Ortiz Ocaña, F (2008): "*Structure Sensor convierte la cámara del iPad en un sensor 3D"* http://aumentada.blogspot.com.es/2014/01/structure-sensor-convierte-la-camara.html consultado en julio del 2014.

[20]. Xataka (2013): "*Kinect para Windows ya está listo para escanear cuerpos y objetos en 3D*" http://www.xataka.com/consolas-y-videojuegos/kinect-para-windows-ya-esta-listo-para-escanear-cuerpos-y-objetos-en-3d consultado en julio del 2014.

[21]. Leap Motion Launches Its Motion-Controlled App Store http://mashable.com/2013/06/24/leap-motion-airspace/ consultado en julio del 2014.

[22]. XBOX 360 y PS3, a la caza de la Wii http://www.nosplay.com/articulos/xbox-360-y-ps3-a-la-caza-de-la-wii consultado en julio del 2014.

[23]. Wii Remote controller (Wiimote) http://www.engadget.com/products/nintendo/wii/remote-controller/specs/ consultado en julio del 2014.